Reduction Techniques for Model Checking Real-Time Rewrite Theories

Supervision:	Carlos Olarte - olarte@lipn.univ-paris13.fr Kaïs Klai - klai@lipn.univ-paris13.fr Jaime Arias - arias@lipn.univ-paris13.fr
Location:	LIPN, CNRS UMR 7030, USPN.
Duration:	3 years
Financial Support:	Institut Galilée

1 Context

Nowadays, concurrent systems (*i.e.* processes that interact with each other) are ubiquitous in many domains and applications, from biological systems to cloud services. In general, concurrent systems exhibit complex forms of interaction, not only between their internal components, but also with their environment. Giving these systems rigorous foundations is a serious challenge for computer science. We need both (1) to accurately capture the behaviour of the system under study; and (2) to provide reasoning techniques for the verification of system properties.

Rewriting Logic (RL) [18] is a flexible semantic framework whose unit of specification is a rewrite theory $\langle\Sigma, E, R\rangle$. The *equational theory* $\langle\Sigma, E\rangle$ — where Σ is a signature and E is a set of equations — defines system states as algebraic data types. The *dynamics* of the system is specified by the set of rewriting rules R. The flexibility of RL for the specification of concurrent systems has allowed the modeling and verification of systems in different domains [18, 7].

Rewiring modulo SMT [3] is a technique aiming at the specification of open systems in RL. A decidable built-in equational theory is assumed, and the satisfability of formulas in this theory is delegated to an SMT solver. Symbolic analysis can be therefore performed, where terms in the rewrite theory coupled with SMT constraints represent a possibly infinite set of concrete states in a compact way. In recent works [12, 11, 13], we have shown how rewriting modulo SMT is a valuable tool for the analysis of real-time rewrite theories including models for parametric timed automata and time Petri nets. We have provided analysis methods beyond the state of the art tools such as Imitator [1] and Roméo [17], including for instance the verification of (timed) systems following a strategy.

Model-checking (MC) is a well-known formal verification technique. It is based on an automatic procedure that takes a model of a system and a formula expressing a temporal property, and decides whether the system satisfies the property. MC relies on an exhaustive exploration of the state space of the system and, as a consequence, suffers from the problem of state space explosion [20].

The *Symbolic Observation Graph* (SOG) [10, 15] is an abstraction technique to tackle the state explosion problem in MC. Guided by the atomic propositions involved in the formula to be verified, it generates a graph which aggregates into symbolic meta-states the states which are homogeneous with respect to the formula to be verified. These sets of states are encoded and managed symbolically using decision diagram data structures.

In the context of RL, methods similar to SOG have been studied in [9]. Unlike [9], where the approach only concerns state-based properties, the construction of the SOG can be driven by event- or action-based properties [10], or state properties [15]. Moreover, SOG has parallel [4] and hybrid [14] verification tools, which is not the case for the current version of *Maude* [5], a language and a system supporting RL.

2 PhD Project

The PhD student will explore: (1) the state-space reduction provided by SOG for the verification of concurrent and real-time systems specified in RL; and (2) develop symbolic (in the sense of rewriting with SMT formulas) model checking methods for automatically verifying real-time and hybrid systems w.r.t. requirements defined using popular property specification logics, such as timed and untimed LTL and CTL, and/or signal temporal logic (STL).

For the objective (1), the PhD should explore a theory transformation (*e.g.* [11, 12]), inspired by the SOG construction [10, 15]. We expect the student to reconcile the SOG approach with the state reduction techniques for rewrite theories in [9]. Coherently combining these techniques may help in further reducing the state space in verification tasks using RL. These theoretical results must be followed by implementation of the proposed analysis methods in Maude [5]. We also expect the student to provide interfaces between the unified Model Checking tool for Maude (`umaudesmc` [19, 6]) and tools based on the SOG (PMC-SOG[1]), thus aiming at equiping Maude with more advanced model checking capabilities.

For the objective (2), we expect the PhD student to develop symbolic model checking techniques for rewriting modulo SMT theories. The specification languages may include untimed linear temporal logic (LTL), and preferably also some timed temporal logic, such as timed CTL or STL. The appropriate model checkers will be implemented and available through a tool like Real-Time Maude [16] and benchmarked with the large set of case studies available in [12, 11, 13]. We expect the student to explore the use of the reduction techniques in (1) in the context of rewriting modulo SMT theories. We also foreseen the exploration of other techniques such as narrowing (rewriting with logical variables) [2, 8] to cope with larger verification tasks in RL models.

The PhD student will collaborate with the members of the verification team at LIPN, members of FST (Tunis, Tunisia), Javeriana Universtiy (Cali, Colombia), POSTECH (Pohang, South Korea) and Oslo University (Norway) in the context of the research projects PISTACHE (lead by Arias, Klai and Olarte) and the NATO project SymSafe (lead by Olarte).

References

[1] É. André. IMITATOR 3: Synthesis of timing parameters beyond decidability. In A. Silva and K. R. M. Leino, editors, *CAV 2021*, volume 12759 of *LNCS*, pages 552–565. Springer, 2021.

[2] K. Bae and J. Meseguer. Infinite-state model checking of LTLR formulas using narrowing. In S. Escobar, editor, *WRLA 2014*, volume 8663 of *LNCS*, pages 113–129. Springer, 2014.

[3] Camilo Rocha, J. Meseguer, and C. A. Muñoz. Rewriting modulo SMT and open system analysis. *J. Log. Algebraic Methods Program.*, 86(1):269–297, 2017.

[4] Chiheb Ameur Abid, Kaïs Klai, Jaime Arias, and H. Ouni. Sog-based multi-core LTL model checking. In *ISPA 2020*, pages 9–17. IEEE, 2020.

[5] M. Clavel, F. Durán, S. Eker, P. Lincoln, N. Martí-Oliet, J. Meseguer, and C. L. Talcott, editors. *All About Maude - A High-Performance Logical Framework, How to Specify, Program and Verify Systems in Rewriting Logic*, volume 4350 of *LNCS*. Springer, 2007.

[6] F. Durán, Camilo Rocha, and J. M. Álvarez. Tool interoperability in the maude formal environment. In *CALCO 2011*, volume 6859 of *LNCS*, pages 400–406. Springer, 2011.

[7] F. Durán, S. Eker, S. Escobar, N. Martí-Oliet, J. Meseguer, R. Rubio, and C. L. Talcott. Programming and symbolic computation in maude. *J. Log. Algebraic Methods Program.*, 110, 2020.

[8] S. Escobar, R. López-Rueda, and J. Sapiña. Symbolic analysis by using folding narrowing with irreducibility and SMT constraints. In C. Artho and P. C. Ölveczky, editors, *FTSCS 2023*, pages 14–25. ACM, 2023.

[1] https://sites.lipn.univ-paris13.fr/websites/pmc-sog

[9] A. Farzan and J. Meseguer. State space reduction of rewrite theories using invisible transitions. In *AMAST 2006*, volume 4019 of *LNCS*, pages 142–157. Springer, 2006.

[10] S. Haddad, J. Ilié, and Kais Klai. Design and evaluation of a symbolic and abstraction-based model checker. In *ATVA 2004*, volume 3299 of *LNCS*, pages 196–210. Springer, 2004.

[11] Jaime Arias, Kyungmin Bae, Carlos Olarte, P. C. Ölveczky, L. Petrucci, and F. Rømming. Rewriting logic semantics and symbolic analysis for parametric timed automata. In *FTSCS 2022*, pages 3–15. ACM, 2022.

[12] Jaime Arias, Kyungmin Bae, Carlos Olarte, P. C. Ölveczky, L. Petrucci, and F. Rømming. Symbolic analysis and parameter synthesis for time petri nets using maude and SMT solving. In *PETRI NETS 2023*, volume 13929 of *LNCS*, pages 369–392. Springer, 2023.

[13] Jaime Arias, Kyungmin Bae, Carlos Olarte, Peter Csaba Ölveczky, Laure Petrucci, and F. Rømming. Symbolic analysis and parameter synthesis for networks of parametric timed automata with global variables using Maude and SMT solving. *Science of Computer Programming*, 233, 2024.

[14] Kais Klai, Chiheb Ameur Abid, Jaime Arias, and S. Evangelista. Hybrid parallel model checking of hybrid LTL on hybrid state space representation. In *VECoS 2021, Revised Selected Papers*, volume 13187 of *LNCS*, pages 27–42. Springer, 2021.

[15] Kais Klai and D. Poitrenaud. MC-SOG: an LTL model checker based on symbolic observation graphs. In *PETRI NETS 2008*, volume 5062 of *LNCS*, pages 288–306. Springer, 2008.

[16] D. Lepri, E. Ábrahám, and P. C. Ölveczky. Sound and complete timed CTL model checking of timed kripke structures and real-time rewrite theories. *Sci. Comput. Program.*, 99:128–192, 2015.

[17] D. Lime, O. H. Roux, C. Seidner, and L. Traonouez. Romeo: A parametric model-checker for petri nets with stopwatches. In S. Kowalewski and A. Philippou, editors, *TACAS 2009*, volume 5505 of *LNCS*, pages 54–57. Springer, 2009.

[18] J. Meseguer. Twenty years of rewriting logic. *J. Log. Algebraic Methods Program.*, 81(7-8):721–781, 2012.

[19] R. Rubio, N. Martí-Oliet, I. Pita, and A. Verdejo. Model checking strategy-controlled systems in rewriting logic. *Autom. Softw. Eng.*, 29(1):7, 2022.

[20] A. Valmari. The state explosion problem. In W. Reisig and G. Rozenberg, editors, *ACPN 2019*, volume 1491 of *LNCS*, pages 429–528. Springer, 1996.